__CTFrypto__

•••

slash

by meowkb.tw

0x00 歷(ㄔㄠ)史(ㄉㄨㄛ)故(ㄍㄟˋ)事(ㄕㄨㄚˋ)

Crypto?

- Kryptos隱藏 + Graphein書寫 (希臘文)

- 有意義的文字 —--混淆----> 無意義的文字

- privacy、integrity、non-repudiation
- 私密性、完整性、不可否認性

用途

- 加密
  - 驗證
  - 隱藏
- Digital Signature 數位簽章
  - 不可否認性
  - 證明文件是你ㄉ
- 保護你的資料

名詞

- 加密 Encrypt：指將明文經過某種程序轉換成密文，該程序稱為加密
- 解密 Decrypt：指將密文經過某種程序轉換成明文，該程序稱為解密
- 明文 Plaintext：加密前的訊息
- 密文 Ciphertext：加密後的訊息
- 演算法 Algorithm：解決複雜問題的程序
- 密碼學演算法：做與密碼學相關程序（如加密、解密、簽章...）的演算法
- 金鑰 / 密鑰 Key：加解密時所使用的「鑰匙」

how to 怕Crypto

- 能用就用**線上工具**
  - rapidtables
- 腳本
  - python
  - cpp
  - java
  - shell script
    - bash shell...
- 通靈

by meowkb.tw

0x01 數字系統

數字系統

- Base
- 補數　(2ㄉ補數 -> 正負號)
- 2的10次方= 1024
    - 1Byte=8bit 、1GB=1024MB、1KB=1024Byte

符號									
0	1	1	1	1	1	1	1	=	127
0	0	0	0	0	0	1	0	=	2
0	0	0	0	0	0	0	1	=	1
0	0	0	0	0	0	0	0	=	0
1	1	1	1	1	1	1	1	=	−1
1	1	1	1	1	1	1	0	=	−2
1	0	0	0	0	0	0	1	=	−127
1	0	0	0	0	0	0	0	=	−128

CS常用進制

- 10進制(Decimal, 簡稱DEC):$178_{(10)}$　PS: 通常省略基數, 直接寫成178
- **2進制**(Binary, 簡稱BIN):$10110010_{(2)}$、**0b**10110010
- 8進制(Octal, 簡稱OCT):$262_{(8)}$
- **16進制**(Hexadecimal, 簡稱HEX):$B2_{(16)}$、0xB2　PS: 數字符號0~9、A~F

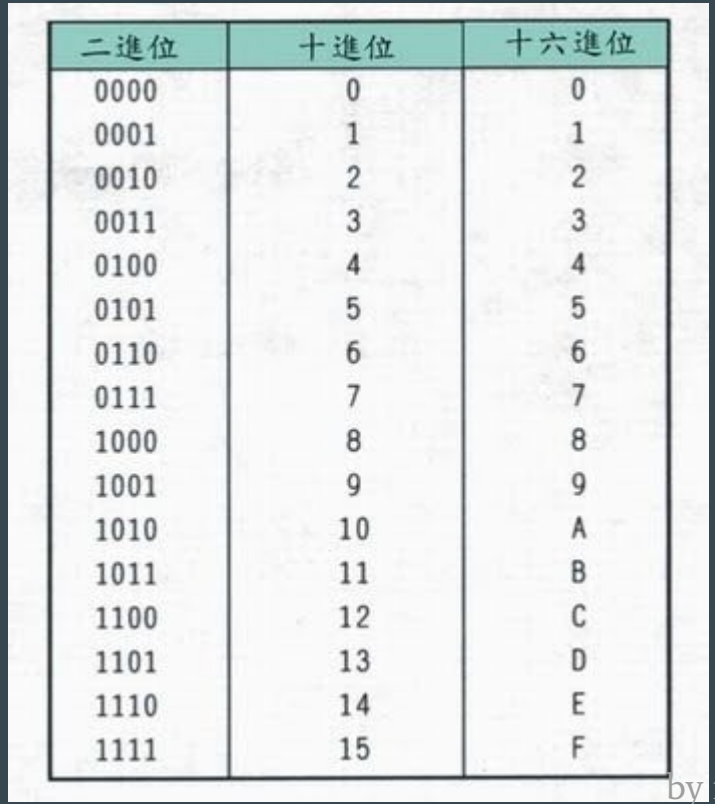
2進制

- Binary　簡稱BIN
- .exe
- 電子電路 半導體
- bitㄉ基礎

十進位制	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
二進位制	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111
八進位制	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
十六進位制	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

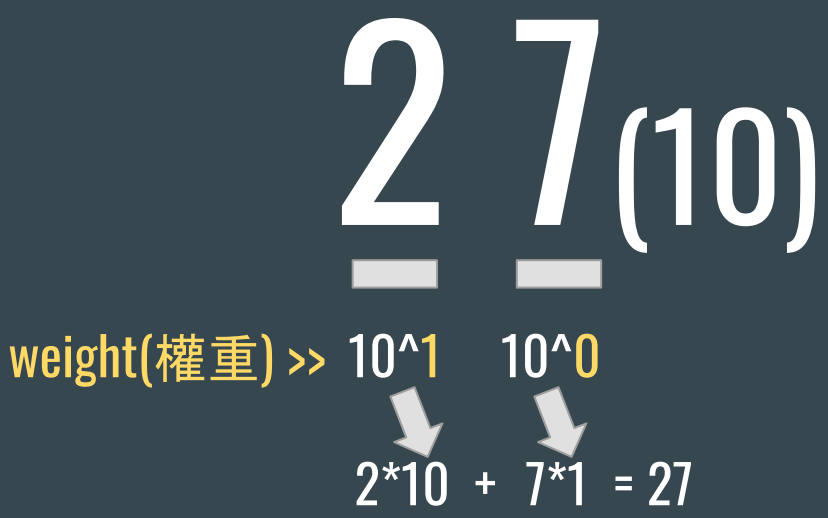
16進制

- HEX
- file signature、組語
- 0x 、&B
- 非常非常重要
- ex: hex to ascii



二進位	十進位	十六進位
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

weight



weight

byte

0 1 0 0 0 0 0 0

2^4=16

2^4=16

2^8=256

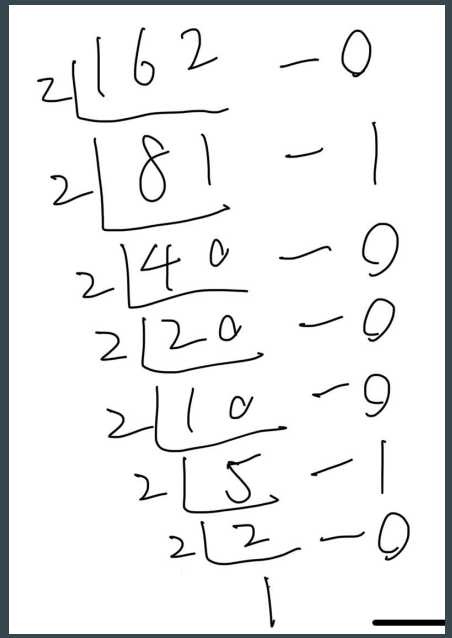
Q: A2=?

人腦:

A=10

$10*16 + 2*1 = 162_{(10)}$

162用短除法⬇️



Online Tools

RapidTables

Home › Conversion › Number conversion › Hex to binary

Hex to Binary converter

From: Hexadecimal — To: Binary

Enter hex number: A2 (16)

= Convert | × Reset | ⇄ Swap

Binary number: 10100010 (2)

Decimal number: 162 (10)

一行python

```
>>> print(bin(0xA2))
0b10100010
```

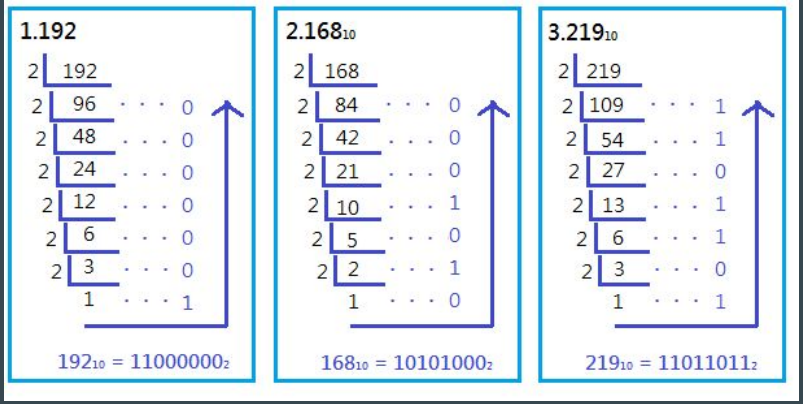
weight

weight(權重) >> 2^4 2^3 2^2 2^1 2^0

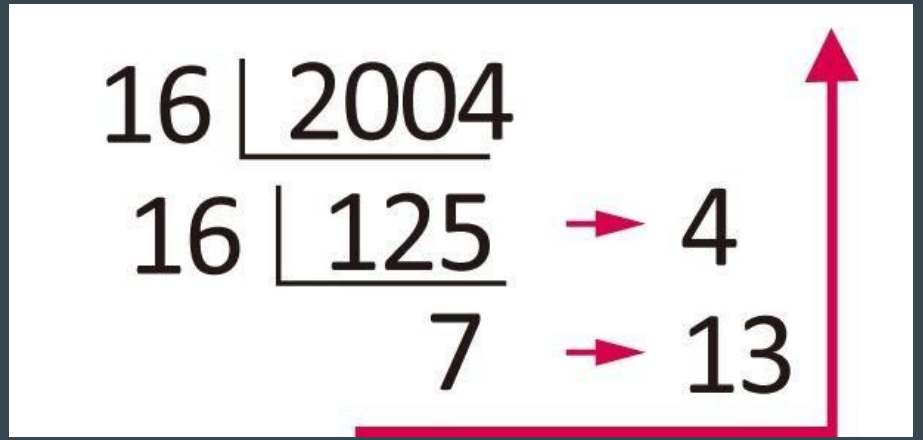
1*16 + 0*8 + 1*4 + 1*2 + 0*1 = 22

數字系統轉換

- Base 10 to Base 2



- Base 10 to Base 16



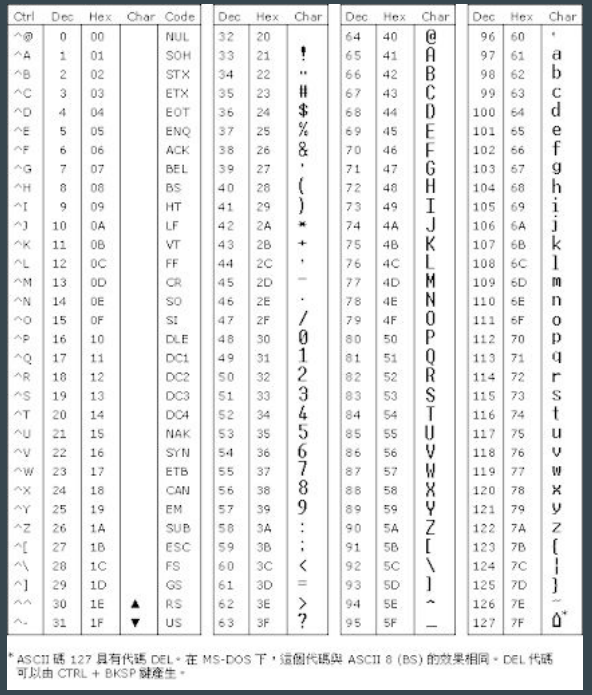
0x03 編碼

概念

- encode & decode 編碼解碼
- 把文字轉成機器看得懂的
- ≠加、解密

常見常用

- 摩斯密碼
- **ASCII**
  - 大小寫英文、數字、常用符號
  - CR、LF
- Unicode (UTF-8)
  - 可轉中文
- Url encoding (Percent-encoding



Ctrl	Dec	Hex	Char	Code
^@	0	00		NUL
^A	1	01		SOH
^B	2	02		STX
^C	3	03		ETX
^D	4	04		EOT
^E	5	05		ENQ
^F	6	06		ACK
^G	7	07		BEL
^H	8	08		BS
^I	9	09		HT
^J	10	0A		LF
^K	11	0B		VT
^L	12	0C		FF
^M	13	0D		CR
^N	14	0E		SO
^O	15	0F		SI
^P	16	10		DLE
^Q	17	11		DC1
^R	18	12		DC2
^S	19	13		DC3
^T	20	14		DC4
^U	21	15		NAK
^V	22	16		SYN
^W	23	17		ETB
^X	24	18		CAN
^Y	25	19		EM
^Z	26	1A		SUB
^[	27	1B		ESC
^\	28	1C		FS
^]	29	1D		GS
^^	30	1E	▲	RS
^-	31	1F	▼	US

Dec	Hex	Char
32	20	
33	21	!
34	22	"
35	23	#
36	24	$
37	25	%
38	26	&
39	27	'
40	28	(
41	29	)
42	2A	*
43	2B	+
44	2C	,
45	2D	-
46	2E	.
47	2F	/
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9
58	3A	:
59	3B	;
60	3C	<
61	3D	=
62	3E	>
63	3F	?

Dec	Hex	Char
64	40	@
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	\
93	5D	]
94	5E	^
95	5F	_

Dec	Hex	Char
96	60	`
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	|
125	7D	}
126	7E	~
127	7F	⌂*

* ASCII 碼 127 具有代碼 DEL。在 MS-DOS 下，這個代碼與 ASCII 8 (BS) 的效果相同。DEL 代碼可以由 CTRL + BKSP 鍵產生。

Base

- Base16/32/64/85
  - 大小寫英文、數字、+/
  - 3個字節一組，用=補足
- 圖片轉base64
- 常用ㄅ通靈編碼

索引	对应字符	索引	对应字符	索引	对应字符	索引	对应字符
0	A	17	R	34	i	51	z
1	B	18	S	35	j	52	0
2	C	19	T	36	k	53	1
3	D	20	U	37	l	54	2
4	E	21	V	38	m	55	3
5	F	22	W	39	n	56	4
6	G	23	X	40	o	57	5
7	H	24	Y	41	p	58	6
8	I	25	Z	42	q	59	7
9	J	26	a	43	r	60	8
10	K	27	b	44	s	61	9
11	L	28	c	45	t	62	+
12	M	29	d	46	u	63	/
13	N	30	e	47	v		
14	O	31	f	48	w		
15	P	32	g	49	x		
16	Q	33	h	50	y		

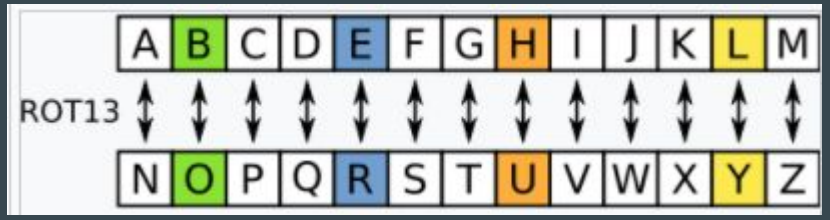
光速判斷Base

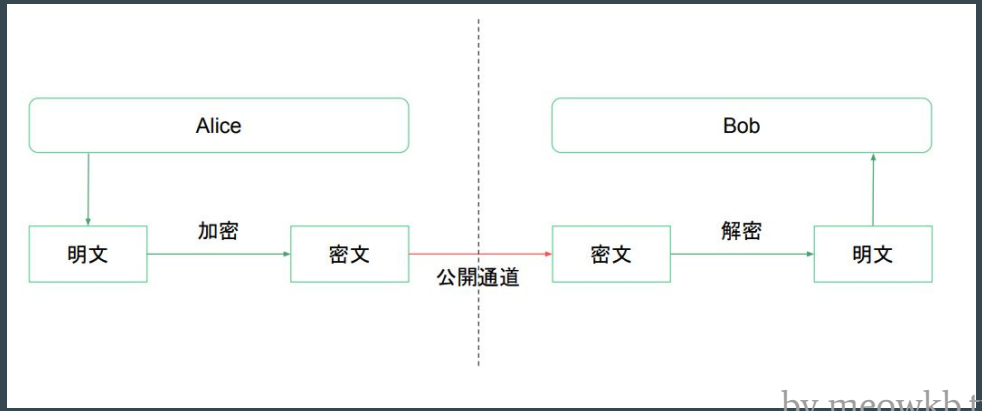
- Base16
  - 跟hex的字元一樣
  - 不會有=
- Base32
  - 只有大寫
  - 至多六個=
- Base64
  - 有小寫 至多2個=
- Base85
  - 用臉在鍵盤上滾一圈

0x04 古典密碼學

easy古典加密

- 替換Substitution
  - ROT-13
  - Caeser Cipher凱薩密碼
  - 維吉尼亞密碼
- 位移Shift
  - 四方密碼
  - 中國密碼





0x05 現代密碼學

一點點ㄉ大綱

- 非對稱式加密 (2 Key
- 對稱 (1 Key
- 雜湊Hash (0 Key

0x06 分組密碼

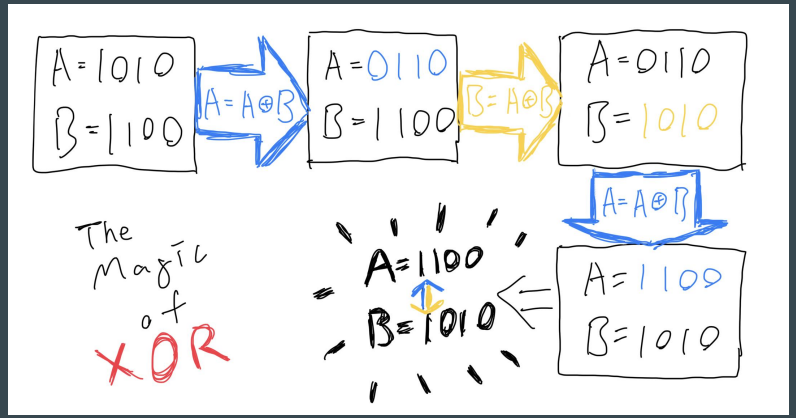
分組密碼

- aka區塊加密
- 塊的概念
- 狂用XOR (可逆性
- ECB、CBC、CFB、OFB

A	B	A **XOR** B
0	0	0
0	1	1
1	0	1
1	1	0

分組密碼_XOR

- ex:不用第三個變數swap value



A	B	A **XOR** B
0	0	0
0	1	1
1	0	1
1	1	0

https://github.com/KvN1027/swap_value_without_third

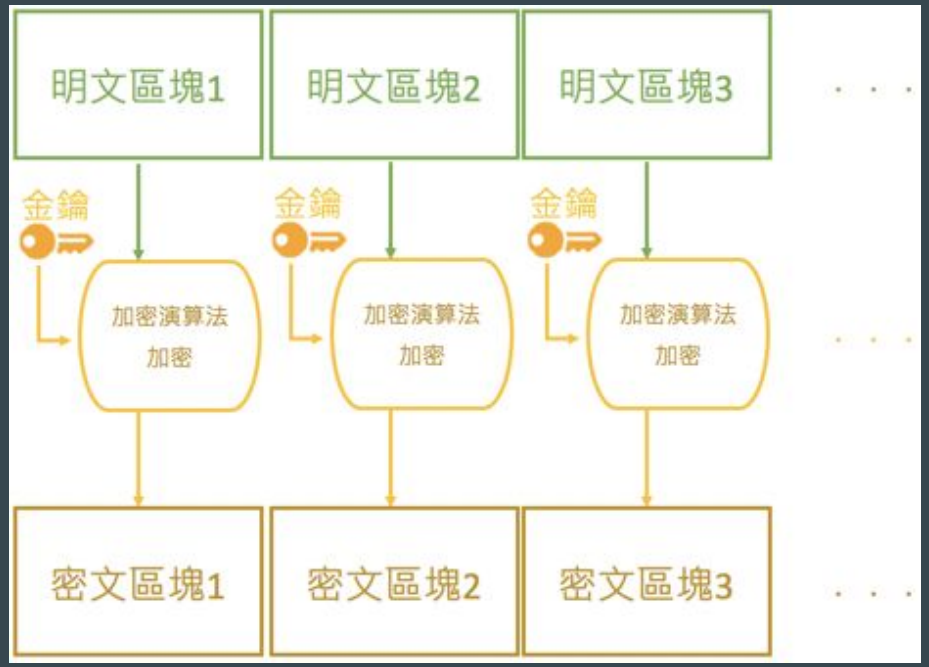
分組密碼_XOR

A	B	A **XOR** B	A	B	A **XOR** B
0	0	0	0	0	0
0	1	1	0	1	1
1	0	1	1	0	1
1	1	0	1	1	0

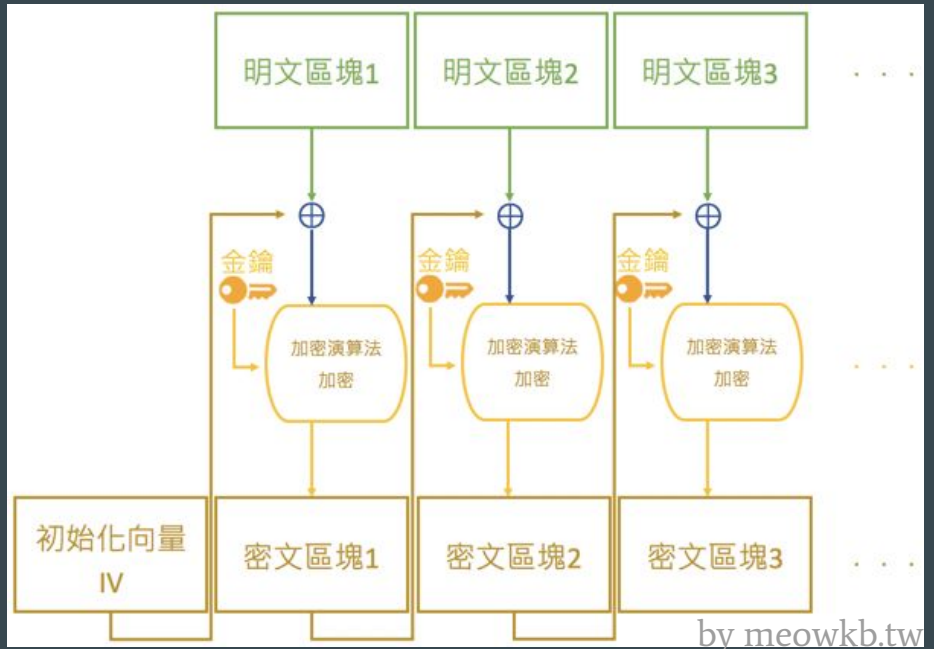
https://github.com/KvN1027/swap_value_without_third

分組密碼_ECB、CBC

- EBC (一對一

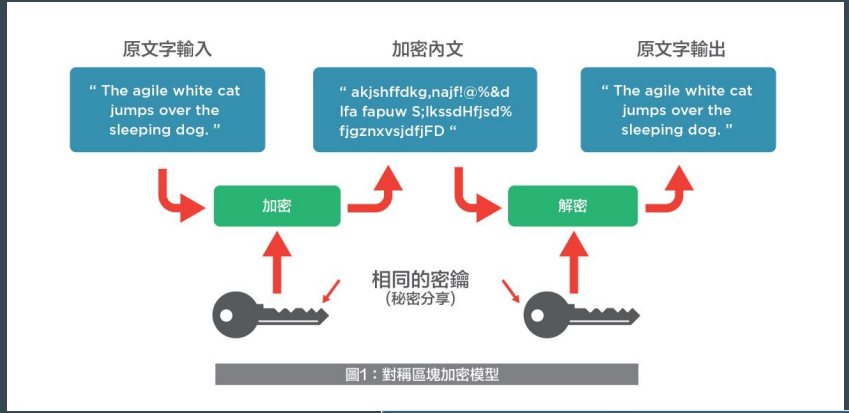


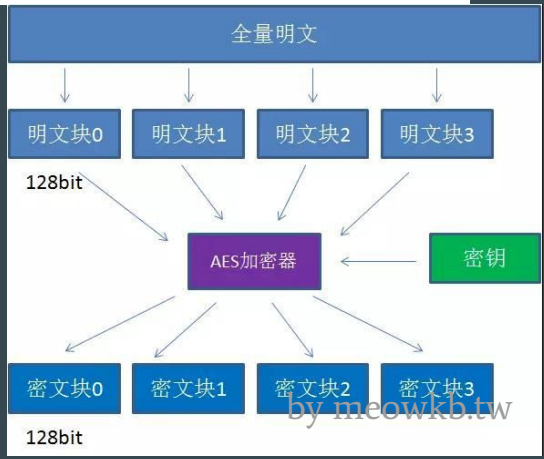
- CBC (一連一



AES

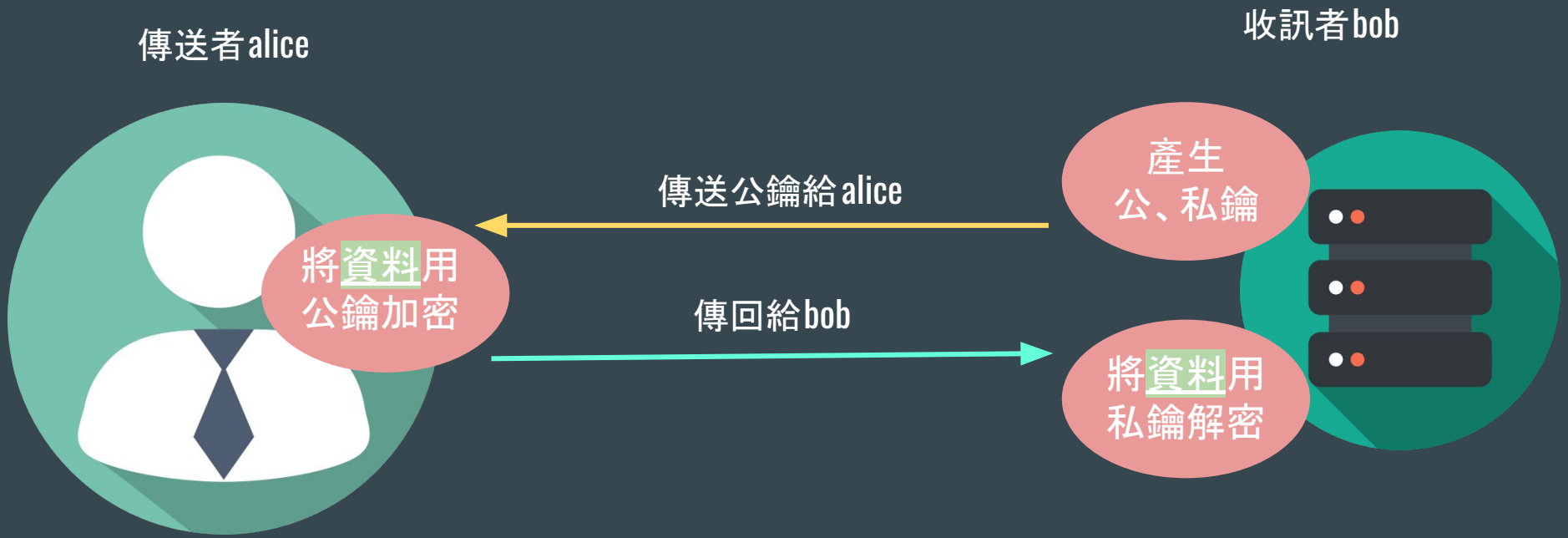
- 繼承DES，不過更安全
- 128、192、256位
- ECB、CBC、CTR、CFB、OFB
- PADDING (補位
  - PKCS5Padding(16位)
    - {1,2,3,4,5,a,b,c,d,e}>{1,2,3,4,5,a,b,c,d,e,6,6,6,6,6,6}
  - ISO10126Padding(16位)
    - {1,2,3,4,5,a,b,c,d,e}>{1,2,3,4,5,a,b,c,d,e,w,@,3,4,C,6}





0x06 非對稱式加密

非對稱式加密流程(a欲傳遞訊息給b)



※公鑰加密過的資料，由私鑰解密

RSA

- hint: RSA分別是三個人的姓氏開頭
- 極大整數的因式分解
- 模指數運算、歐拉函數、同餘

RSA_初始運算

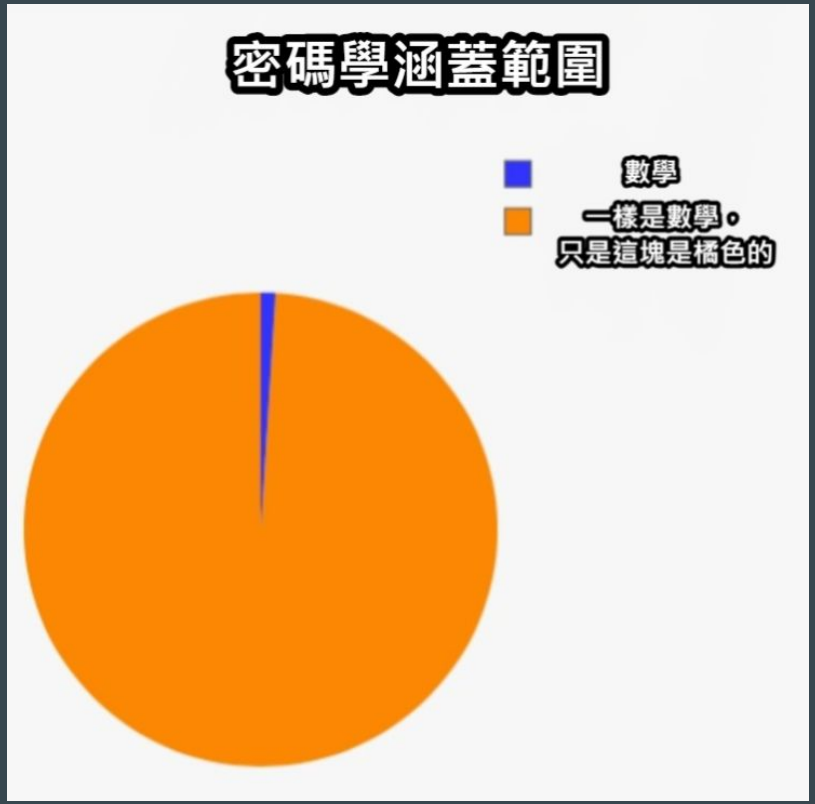
1. 選出兩個較大的質數 **p** 和 **q**
2. 計算兩個質數的乘積 **n = p × q**
3. 計算出小於 n 且與 n 互質的整數個數 **φ(n)= (p - 1)(q - 1)**
4. 選出一個整數 **e**（拿來當做公鑰）
    a. 選擇條件
        i. **1 < e < φ(n)**
        ii. **e** 與 **φ(n)** 互質(means沒有共同因子)
5. 計算 **d** (私鑰)
    a. **d** 與 **e** 的關係
        i. **e x d / φ(n)**餘數為 1　　(ex:**φ(n)=20 -> e=3 d=7 -> 3*7/20=1**)
        ii. 因此 **d = e - 1 mod**
6. 可得
    a. 公鑰 **KU = {e, n}**。
    b. 私鑰 **KR = {d, n}**。

RSA_加密流程

1. 先將明文轉為數字(m), 轉換成數字的方法根據編碼
2. $m^e \pmod n = c$

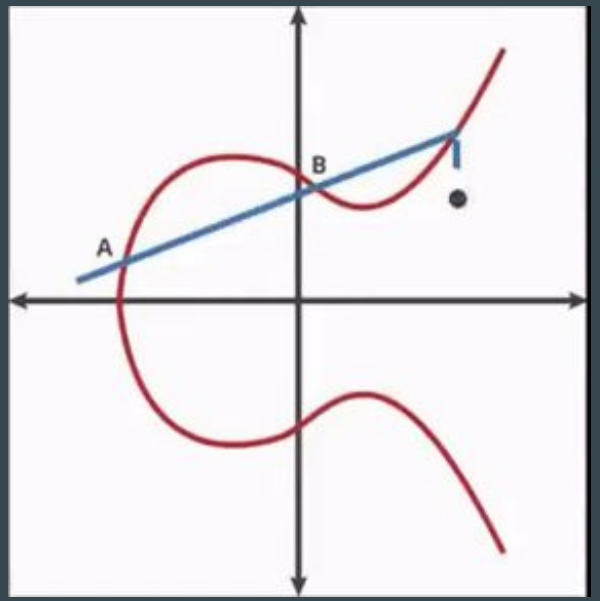
e為公鑰　n為整數(p*q)　c為轉換後的密文

！！！ 大量數學湧入中 ！！！



ECC橢圓曲線加密

- Elliptic Curve Cryptography
- 非對稱式加密
- 公認相同key長度內最強加密
- ex:中國身分證、比特幣
- ECCTooL



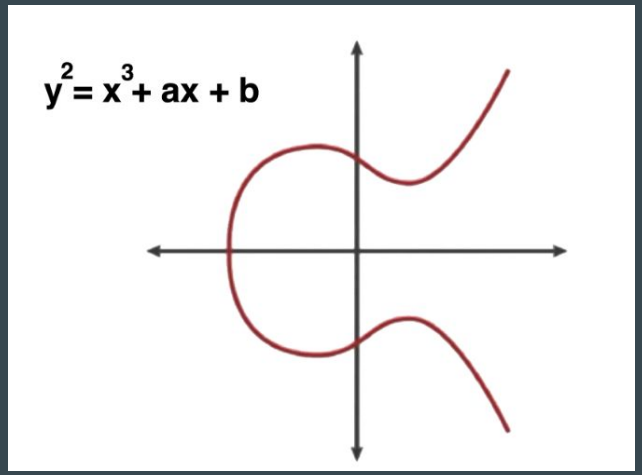
ECC橢圓曲線加密

RSA vs ECC 比較

PKI Algorithm	RSA	ECC
Key Size	Security: 280 @ 1024-bits	Security: 280 @160-bits
	Security: 2112 @ 2048-bits	Security: 2112 @224-bits
	Security: 2128 @ 3072-bits	Security: 2128 @ 256-bits
	Security: 2:92 @ 7680-bits	Security: 2192 @386-bits
	Security: 2256 @ 15360-bits	Security: 2256 @512-bits
安全基礎	大數因數分解	EC橢圓曲線上離散對數
優點	演算法容易說明,且同時可用做加解密	運算速度快,簽章長度較小
缺點	運算速度慢,簽章長度較大	理論較難理解,且實作技術較為複雜

ECC橢圓曲線加密

- 最常用的橢圓公式



- 有限域

伽羅瓦域 [編輯]

對橢圓曲線來說最流行的有限域是以素數為模的整數域（參見模運算）$GF(p)$，或是特徵為2的伽羅瓦域 GF（2^m）。後者在專門的硬體實現上計算更為有效，而前者通常在通用處理器上更為有效。專利的問題也是相關的，一些其他素數的伽羅瓦域的大小和能力也已經提出了，但被密碼專家認為有一點問題。

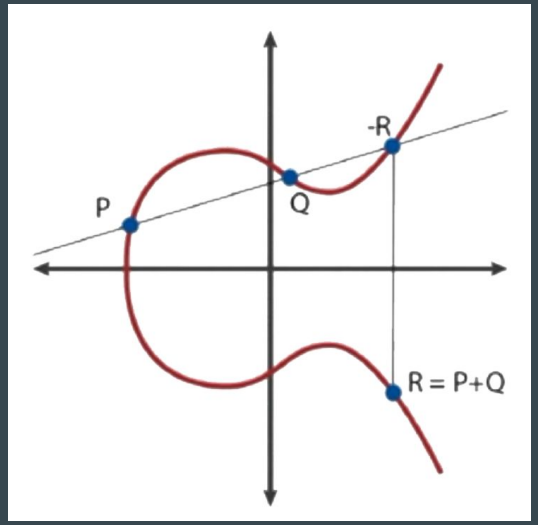
給定一條橢圓曲線E以及一個域$GF(q)$，考慮具有(x, y)形式有理數點$E(q)$的阿貝爾群，其中x和y都在$GF(q)$中並且定義在這條曲線上的群運算"+"（運算"+"在條目橢圓曲線中描述）。然後定義第二個運算"*"| Z×$E(q)$− > $E(q)$：如果P是$E(q)$上的某個點，那麼定義$2 * P = P + P, 3 * P = 2 * P + P = P + P + P$等等。計對給定整數和k，$j * (k * P) = (jk) * P = k * (j * P)$。橢圓曲線離散對數問題（ECDLP）就是給定點P和Q，確定整數k使$k * P = Q$。-- 一般認為在一個有限域乘法群上的離散對數問題（DLP）和橢圓曲線上的離散對數問題（ECDLP）並不等價；ECDLP比DLP要困難的多。

在密碼的使用上，會選擇曲線$E(q)$和其中一個特定的基點G，並且公開這些資料。會再選擇一個隨機整數k作為私鑰；公布值為$P = k * G$的公鑰（注意假設的ECDLP困難性意味著k很難從P中確定）。如果Alice和Bob有私鑰k_A和k_B，公鑰是P_A和P_B，那麼Alice能計算k_A*P_B=(k_A*k_B)*G；Bob能計算同樣的值k_B*P_A=(k_B*k_A)*G。

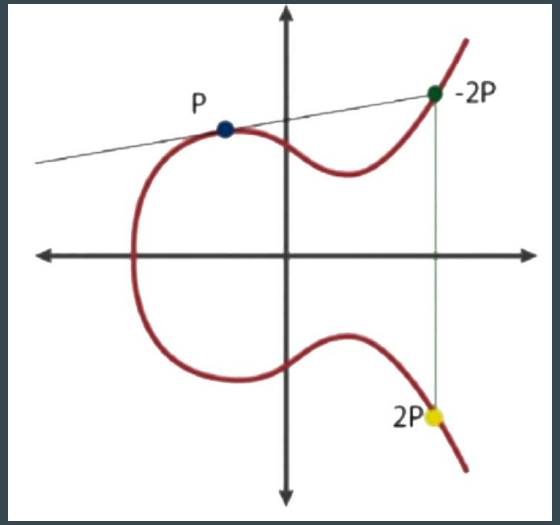
這允許一個「秘密」值的建立，這樣Alice和Bob能很容易地計算出，但任何的第三方卻很難得到。另外，Bob在處理期間不會獲得任何關於k_A的新知識，因此Alice的私鑰仍然是私有的。

ECC橢圓曲線加密　　P、Q兩點位置關係

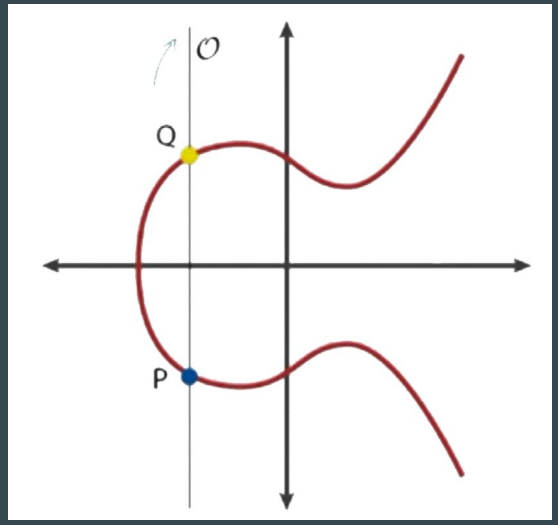
- 加法



- 乘法



- 無窮遠點

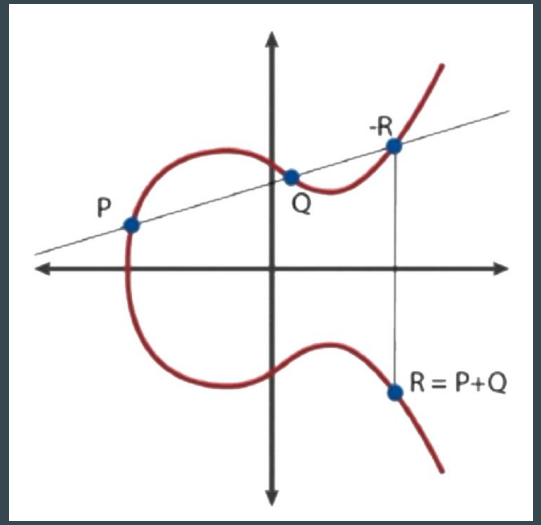


by meowkb.tw

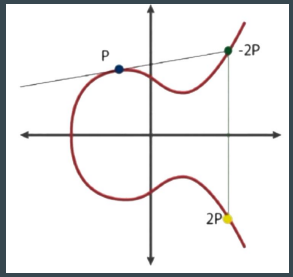
ECC橢圓曲線加密　　P、Q兩點define

1. 橢圓曲線方程式為**y2=x3+ax+b**
2. 此曲線剛好對稱於**x軸(y=0)**這條直線
3. 參數a及b必需滿足**4a3+27b2≠0**,才能確保沒有重根, 具有唯一解!
4. 加法單位元素**O**為一無窮遠的點, 並滿足**O=-O**
5. 此加法單位元素亦需滿足: **橢圓曲線上某三點共線其合為O**

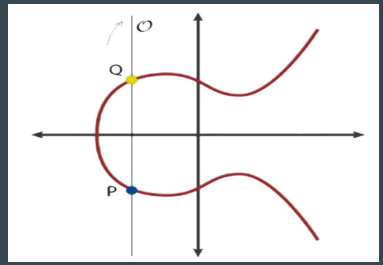
- 加法



- 乘法



- 無窮遠點

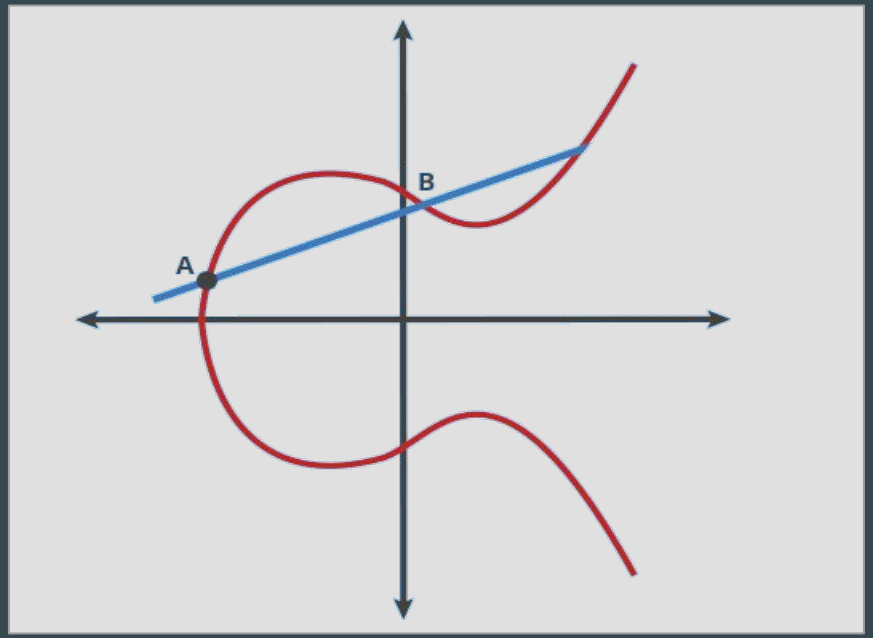


by meowkb.tw

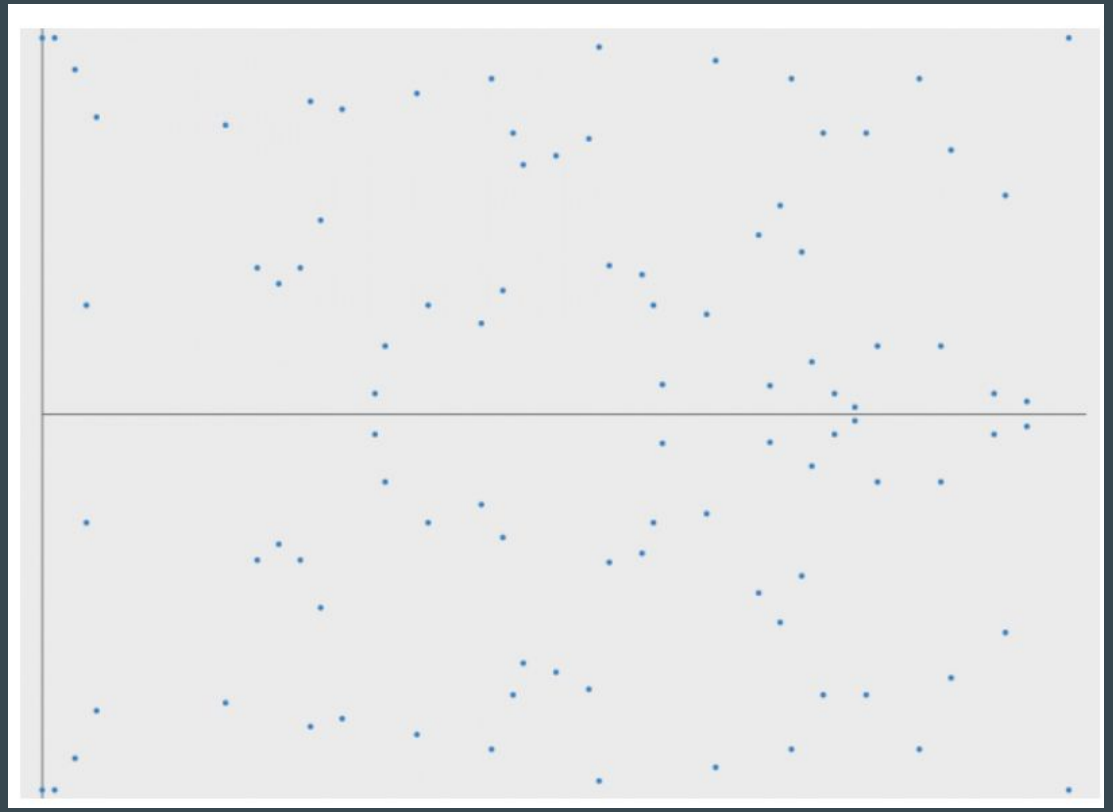
ECC橢圓曲線加密　　P、Q特性

反彈n次(as Private Key) 會得到
一個最終點Q(as Public Key)

n超級大



ECC橢圓曲線加密　　離散化



具體解法　ECC橢圓曲線加密

1、用戶A選定一條適合加密的橢圓曲線Ep(a,b)(如:$y^2=x^3+ax+b$)，並取橢圓曲線上一點　，作為基點G。
2、用戶A選擇一個私有密鑰k，並生成公開密鑰K=kG。
3、用戶A將Ep(a,b)和點K，G傳給用戶B。
4、用戶B接到信息後，將待傳輸的明文編碼到Ep(a,b)上一點M，並產生一個隨機整數r（r<n）。
5、用戶B計算點C1=M+rK；C2=rG。
6、用戶B將C1、C2傳給用戶A。
7、用戶A接到信息後，計算C1-kC2，結果就是點M

0x07 雜湊

雜湊

- hash

```
(Key)                    (hash value)      (stored index)
Joe  → (Hash function) →   4928   mod 5   =   3
Sue  → (Hash function) →   7291   mod 5   =   1
Dan  → (Hash function) →   1539   mod 5   =   4
Nell → (Hash function) →   6276   mod 5   =   1
Ally → (Hash function) →   9143   mod 5   =   3
Bob  → (Hash function) →   5278   mod 5   =   3
```

雜湊

- md5
  - 檔案驗證
- sha0/1/2/3
  - sha2 -> bitcoin

雜湊

- 碰撞攻擊
  - 鴿籠原理
- 彩虹表
  - 先算好固定長度